Observing Resources in the Wild

Graziano Obertelli

The Mayhem Group
University of California, Santa Barbara

May 12, 2004

Grid Computing

- Vision: Applications programs *plugs* into a software environment to draw computational *power* from a dynamically changing pool of resources *(Foster, Kesselman, et al, 1998)*.
- Electrical Power Grid analogy:
  - Power generation facilities
  - Household appliance
  - Scale to national and international levels
- Federated System:
  - Grid users (both power producers and applications consumers) *must* be able to join and leave the Grid at will
  - Local control supersedes global control

The Dark Side of Grid Computing

- Heterogeneity
  - Machines, networks, software, administrative policies all vary
  - and *electrons* are not created equals (CPU cycles, memory and storage space, connectivity)
- Dynamism
  - Loads, performance and availability change with time
- Programmability
  - Complex and dynamically changing system
- Security
- Maintainability

Stormy Weather

- Problem: How can programs extract high performance levels given the resource pool is heterogeneous and dynamically changing?
  - Applications and/or systems must be able to tolerate or mask fluctuating performance of federated grid resources
- Idea: Predict future performance levels and adapt applications to the predictions *on the fly*
  - Predictions must be at the application level
  - Predictions must be made on-line to avoid staleness
  - Delivery of predictions must be fast
- A Solution: The Network Weather Service

Outline

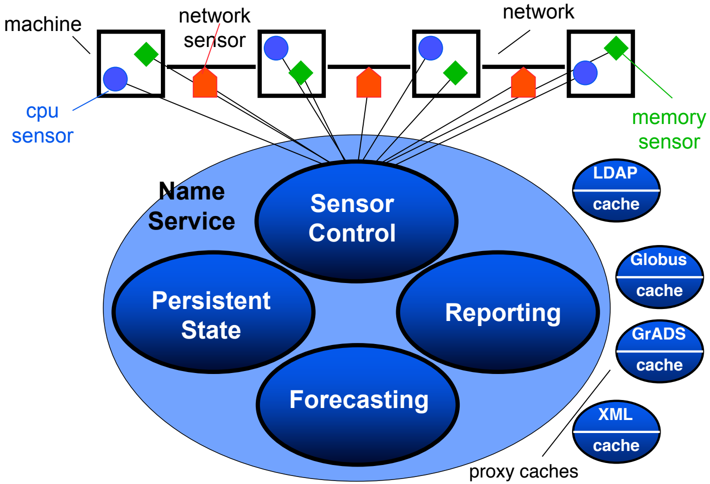
Challenges in Monitoring the Grid

- The monitoring system has to be itself a Grid application.
- Problem: need to take measurements constantly
  - No way to know when the user will make a query
  - Models become stale very quickly
  - Previous history improves accuracy of predictions
- Mort information systems are optimized for query and not update (high frequency dynamics are common)
- The System needs to be:
  - Non-Intrusive: need to control the perturbation on the monitored resources
  - Fast: needs to gather and forecast data from all resources all the time
  - Robust: if the scheduler can't *see* the resource, it can't use it

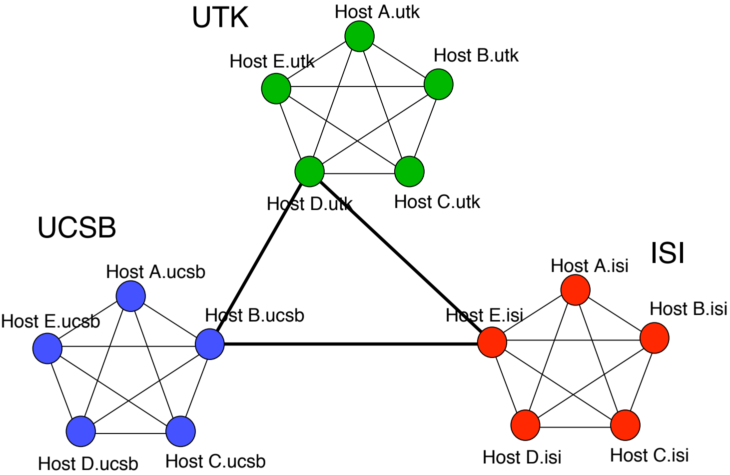
The Network Weather Service

- A distributed, robust and adaptive system that
  - monitors the performance that is available from distributed resources
  - forecasts future performance levels using fast statistical techniques
  - delivers forecasts *on-the-fly* to applications and resource allocators (Globus, Condor, Legion, NetSolve, NINF etc. . .)
- Portable and extensible
- Works at the application level and end-to-end
- Structure the system exploits the delay between data generation and query:
  - Sensors write into local, distributed repositories
  - Proxy caches asynchronously *pull* the data close to the user
  - User forecasting to filter the effects of inconsistent *best effort* world view

Logical Architecture



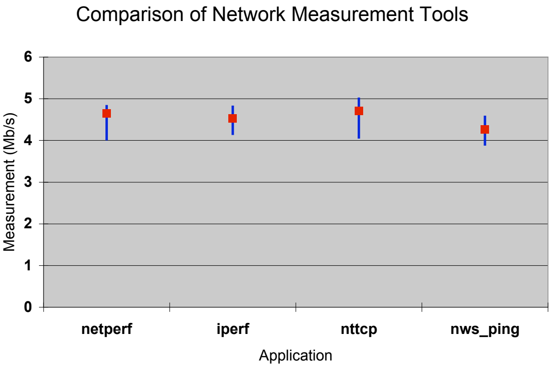
End-to-end Network Measurement



Lesson I: Services vs Tools

- There are a lot of tools: iperf, netperf, pathload, nttcp, ping
  - They are simpler (they monitor only one thing)
  - They are better known (*Iperf is the most accurate end-to-end performance monitoring tool.*)
  - Not designed to gather data continuosly: timeout is ctrl-C
  - Each user runs a separate instance:
    - 23% of the traffic going across Abilene is measuring network performance
- Network Weather *Service*:
  - Allows to tailor the reports per application (cliques are user defined)
  - Measures multiple resources of the same node (ie CPU might be useless without memory)
  - Give fresh *forecasts* of future usage of resources
  - Designed for unattended operations:
    - adaptive timeouts
    - handles network partitions

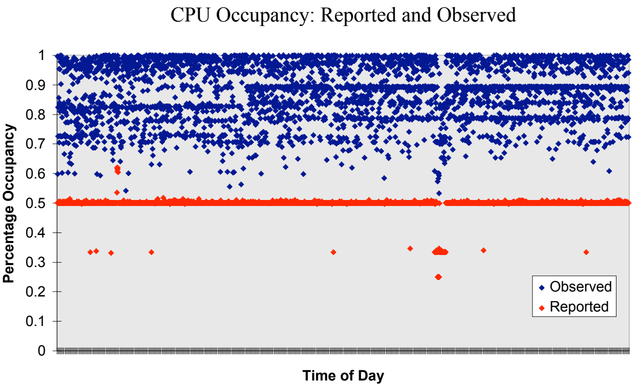
Best of Breeds



Lesson II: Seeing is Not Always Believing

- Measuring seems easy, but knowing what is being measured is hard:
  - Operating systems lie
    - What does load average mean
    - Can Linux really make time go backwards?
  - Network measurements measure the network and not what it delivers:
    - Pathload, pchar, pathchar measure "capacity"
  - Measuring is fun so everybody build a measurement tool: there is much lore
- Measurement error?

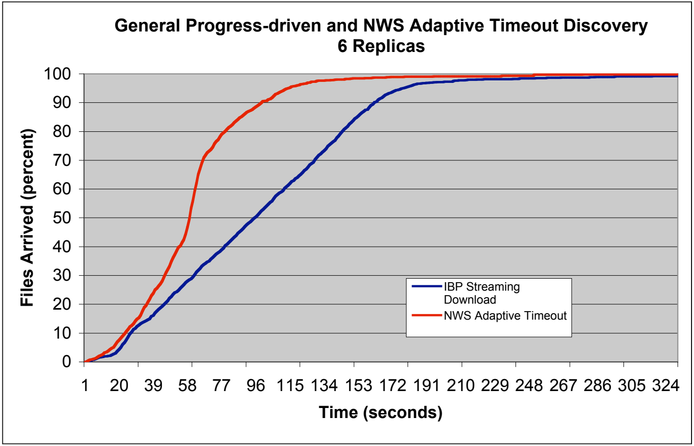
What Does Unix Load Average Measure?



Lesson III:One Size Does Not Fit All

- GridFTP can use parallel TCP sockets for data transfer
  - Useful for *clean* networks with high bandwidth-delay products
  - Several other tools have followed suit
- IBP streaming downloads:
  - Segmented streaming media download protocol
  - Fetches replicated segments in parallel
  - Used a deadline-oriented progress metric to increase parallel fetches of late segments
  - *"No Need for forecasting!"*

Same Robustness Better Performance



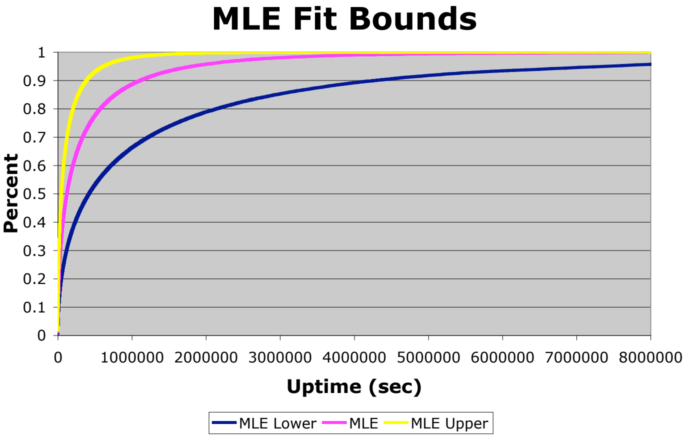
Lesson IV: Real Systems Can Be Predicted

- Can we predict machine availability with quantifiable error?
- UCSB Instructional Computing Labs (6 months)
  - Approximately *85 machines* running Red Hat Linux locate in three separate buildings
  - Open to all Computer Science graduate and undergraduate (approximately 800 users)
  - Power switch is not protected: Students routinely *clean off* competing users or intrusive processes to gain better performance response
- Condor Pool at University of Wisconsin (6 months)
  - *Idle* cycle harvesting system
  - Machine owners specify what *idle* and *busy* mean
  - Approximately 2000 machines
- Darrell Long Internet Study (3 months, 1995)
  - pings rstat daemon
  - Approximately 1200 hosts

Predictions and Confidence

- For Scheduling, predict the quantile for each machine: **For at least how long will a given machine be available with 95% confidence?**
- Requires two estimates:
  - Estimate the 0.05 quantile, call it $Q_{0.05}$ $P(x >= Q_{0.05}) = .95$
  - Estimate the 95% confidence interval on $Q_{0.05}$
- At least 95 times out of 100 the availability should be bigger than lower confidence bound of $Q_{0.05}$

An Example: UCSB



Accurate Predictions and Confidence Values

Data Set	Weibull Method	Resample Method	Binomial Method
CSIL	56.25%	62.5%	87.5%
Condor	95.92%	60.2%	98.9%
Long	57.95%	53.4%	94.3%

- First 20 measurements of each trace use to predict remaining data values
- Picked random future values and recorded the number of time each is successful
- Total correct hit rate for Binomial method is 96.5%

Lessons From the Wild

- A Service and not a Tool:
  - if long-running, robust measurements are needed, simple end-to-end tools are the wrong solution
- Measurements looks easy but understanding measurement is hard:
  - systems should consider measurement error
  - measurements should be taken at the application level
- Don't believe the lore:
  - be adaptable to recognize the different condition and react accordingly
- Good predictions are possible
  - key is to be able to quantify error and confidence

Thanks

- Miron Livny and the Condor group at the University of Wisconsin
- Darrell Long (UCSC) and James Plank (UTK)
- UCSB Facilities Staff
- NSF and DOE
- mayhem group (mayhem@pompone.cs.ucsb.edu):

Rich Wolski

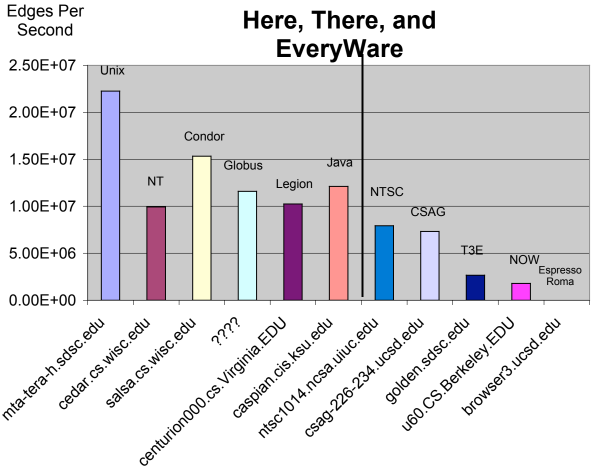
John Brevik	Martin Quinson
Matthew Allen	Dan Nurmi
Todd Bryan	Wahid Chrabakh
Graziano Obertelli	Larry Miller
Ye Wen	Andrew Mutz
Lamia Youseff	Fred Tu

- graziano@cs.ucsb.edu
- http://nws.cs.ucsb.edu

Needs to Run EveryWare

- EveryWare at SC98: The first program to use *all* of the extant Grid infrastructure at the same time.
- The first program to couple the *Tera* and the *NT Supercluster* with a *web browser* in a coffee shop.
- The first *true* Computational Grid program in the visionary sense of the words:
  - cycles are commodities
  - dynamically adaptive and robust
  - completely non dedicated-access
- Portable and easily extensible framework into which applications can be plugged in

The New Resource (Tera MTA) is the Fastest



The Nice Guy

- Original NWS default experiment size was $64K$ bytes (ISDN and 10Mb Ethernet were still in wide use when NWS 1.0 was coded) but the experiment size (and the buffer size and the single message size) is user configurable
- Notable colleagues published a paper containing the following quote: *Although tools such as the Network Weather Service (NWS) measure and predict network bandwidth, a substantial difference in performance can arise between a small NWS probe (lightweight with 64KB size) and an actual file transfer using GridFTP (with tuned TCP buffers and parallelism).*
- Several groups have contacted us complaining about the inability to use large probes

Small Probes Carry Lots of Information